

Scalable Parallel Algorithm of Finite Difference Lattice Boltzmann Method

Lei Xu, Anping Song, Zhixiang Liu and Wu Zhang

Abstract—The lattice Boltzmann method (LBM) has become an attractive and promising approach in computational fluid dynamics (CFD) which can be applied in a wide variety of complex fluid simulations including incompressible flow, compressible flow, microscale gas flow, single flow and multiphase flow. In this paper, the parallel algorithm of D3Q19 finite-difference LBM (FDLBM) is presented to simulate 3D lid-driven cavity flow and flow past a sphere. A 3D domain decomposition method and data exchange strategy are proposed to improve the parallel performance. Details of domain decomposition method are devised based on load balancing on large scale cluster. The numerical results on large scale cluster indicate that the presented algorithm has very good scalability and efficiency. The efficiency can achieve 86.1% on 3072 cores.

Index Terms—finite difference method, lattice Boltzmann method, MPI, domain decomposition method.

I. INTRODUCTION

DURING the past two decades, the standard lattice Boltzmann method (SLBM) has attracted much attention in fluid simulations. It has become an alternative method of CFD for fluid simulations [2], [3], [4]. There are several variations of SLBM including lattice Bhatnagar-Gross-Krook (LBGK) model[5] or single-relaxation-time (SRT) model[6], two-relaxation-time (TRT) model[7], entropic model [8] and multiple-relaxation-time (MRT) model[9]. SLBM describes the evolution of the single particle distribution functions (PDFs) containing two steps: collision and propagation. In the collision step, PDFs are updated locally, and in the propagation step, PDFs are shifted with adjacent lattice nodes. Owing to parallel nature of collision and propagation steps, SLBM is easier implemented on large scale clusters compared with other traditional CFD methods such as finite difference method (FDM), finite element method (FEM) and finite volume method (FVM). However, because the PDFs can only be shifted between adjacent lattice nodes, the Courant-Friedrichs-Lewy (CFL) number of LBM has to be 1. The time step is severely limited by the CFL number which is not suitable for high resolution simulations.

In order to overcome the defects of SLBM, several discrete Boltzmann equation (DBE) methods have been proposed based on direct discretization of DBE in space and time using FDM [10], [11], FEM [12] and FVM [13], [14]. The Lax-Wendroff scheme finite difference LBM is employed in our

Manuscript received February 2, 2017; revised April 18, 2017.

Lei Xu is with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444 China (e-mail: leixushu@t.shu.edu.cn).

Anping Song is associate professor with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444 China (e-mail: apsong@shu.edu.cn).

Zhixiang Liu is with the School of Communication and Information Engineering, Shanghai University, Shanghai, 200444 China (e-mail: lzxshu@shu.edu.cn).

Wu Zhang, the corresponding author, is professor with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444 China (e-mail: wzhang@shu.edu.cn)

study [15], [16] which is an explicit scheme with two order accuracy. Owing to the dimension of the problems treated with the LBM, high computing power and large memory space are required. Therefore, this phenomenon naturally calls for parallel strategies. [18], [19]. In this paper, a detailed parallel algorithm of Lax-Wendroff scheme FDLBM is presented.

The rest of the paper is organized as follows. In Section II, a description of Lax-Wendroff scheme finite difference LBM and boundary condition are given. The detailed parallel strategy is presented in Section III. Next, the numerical results are discussed in Section IV. Finally, the conclusions are summarized in Section V.

II. FINITE DIFFERENCE LATTICE BOLTZMANN METHOD

A. Lax-Wendroff scheme FDLBM

The discrete Boltzmann equation for nearly incompressible single phase flows with a single relaxation time collision operator can be written as

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla f_\alpha = -\frac{1}{\lambda}(f_\alpha - f_\alpha^{eq}), \quad (1)$$

where f_α are the PDFs, t is the time, and λ is the relaxation parameter [15], [16], [17]. For the D3Q19 lattice model (Fig. 1) [20] used in our research, the microscopic velocity set is given as

$$\mathbf{e}_\alpha = \begin{cases} (0, 0, 0), & \alpha=0, \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1), & \alpha=1 \sim 6, \\ (\pm 1, \pm 1, 0), (0, \pm 1, \pm 1), (\pm 1, 0, \pm 1), & \alpha=7 \sim 18. \end{cases} \quad (2)$$

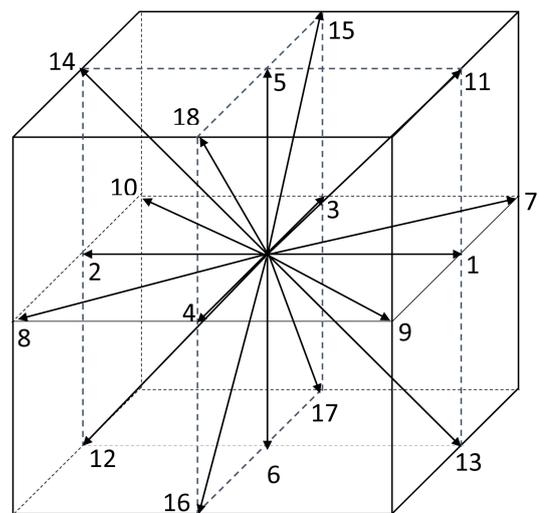


Fig. 1. D3Q19 lattice model

and the equilibrium distribution functions (EDFs) are

$$f_\alpha^{eq} = \rho \omega_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right], \quad (3)$$

where $c_s = \frac{1}{\sqrt{3}}$ is the lattice speed of sound. the weight coefficients are

$$\omega_\alpha = \begin{cases} \frac{1}{3}, & \alpha = 0, \\ \frac{1}{18}, & \alpha = 1 \sim 6, \\ \frac{1}{36}, & \alpha = 7 \sim 18. \end{cases} \quad (4)$$

The density, velocity and pressure can be calculated as follows,

$$\rho = \sum_\alpha f_\alpha, \quad \mathbf{u} = \frac{1}{\rho} \sum_\alpha f_\alpha \mathbf{e}_\alpha, \quad p = \rho c_s^2. \quad (5)$$

According to the reference [15], the DBE can be divided into two separate steps: collision

$$\hat{f}_\alpha = f_\alpha - \frac{f_\alpha - f_\alpha^{eq}}{\tau + 0.5}, \quad (6)$$

and propagation

$$\frac{\partial \hat{f}_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla \hat{f}_\alpha = 0, \quad (7)$$

where $\tau = \frac{\lambda}{\Delta t}$ is the dimensionless relaxation time and Δt is the time step. Because of the Eulerian nature of the propagation step, temporal and spatial discretizations are decoupled so that different spatial resolutions irrespective of the time step are possible. In Eq. (7), central difference is used to discretize the first and second derivatives along characteristic lines, the following Lax-Wendroff scheme with second-order accuracy in time and space can be obtained [15], [16], [21]:

$$\hat{f}_\alpha(\mathbf{x}, t + \Delta t) = -\frac{\sigma(1-\sigma)}{2} \hat{f}_\alpha(\mathbf{x} + \Delta \mathbf{x}_\alpha, t) + \frac{\sigma(1+\sigma)}{2} \hat{f}_\alpha(\mathbf{x} - \Delta \mathbf{x}_\alpha, t) + (1 - \sigma^2) \hat{f}_\alpha(\mathbf{x}, t), \quad (8)$$

where $\sigma = \frac{|\mathbf{e}_\alpha| \Delta t}{\Delta x_\alpha}$ is the CFL number.

It can be seen from Eq. (8), when $\sigma = 1$, the Lax-Wendroff approximation of the propagation is the LBGK propagation in which the spatial resolution is at its finest level.

In the limit of low Mach number, the incompressible Navier-Stokes equations can be obtained through the Chapman-Enskog expansion.

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u}) = -\nabla p + \nabla \cdot [\rho \nu (\nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla)] \end{cases} \quad (9)$$

B. Boundary condition

In our numerical simulations, a non-equilibrium extrapolation method (Fig. 2) is employed for the non-slip static straight wall [22]. It divide the distribution function \hat{f}_α at the boundary into two parts: equilibrium and non-equilibrium:

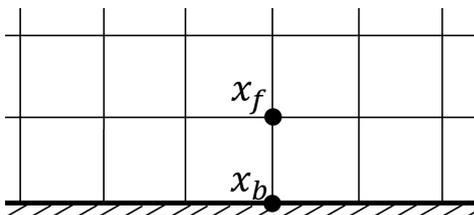


Fig. 2. Non-equilibrium extrapolation method

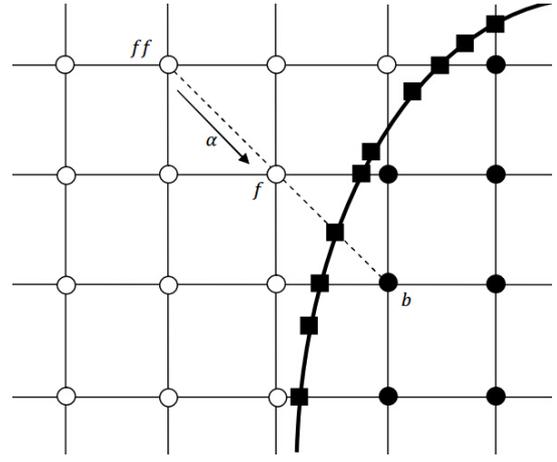


Fig. 3. Illustration of Curved Boundary

$$\hat{f}_\alpha(\mathbf{x}_b, t) = \hat{f}_\alpha^{eq}(\mathbf{x}_b, t) + \hat{f}_\alpha^{neq}(\mathbf{x}_b, t), \quad (10)$$

where the $\hat{f}_\alpha^{eq}(\mathbf{x}_b, t)$ is the equilibrium part and it can be got by Eq. (3). The non-equilibrium part $\hat{f}_\alpha^{neq}(\mathbf{x}_b, t)$ can be approximated by its nearest lattice node \mathbf{x}_f .

$$\hat{f}_\alpha(\mathbf{x}_b, t)^{neq} = \hat{f}_\alpha(\mathbf{x}_f, t) - \hat{f}_\alpha^{eq}(\mathbf{x}_f, t). \quad (11)$$

In our present work, a unified boundary treatment for curved boundary is adopted to model non-slip curved boundary condition [23]. In Figure 3, the solid and hollow circles are the solid and fluid nodes respectively. A curved wall separates the solid nodes from the fluid nodes. \mathbf{x}_f is noted as lattice node on the fluid side of the boundary and \mathbf{x}_b on the solid side. \mathbf{x}_{ff} is the adjacent fluid node of \mathbf{x}_f . The filled small rhombus on the boundary wall \mathbf{x}_w is the intersections of the wall with various lattice links. The boundary velocity at \mathbf{x}_w is noted as \mathbf{u}_w . The curved boundary scheme can be written as

$$f_\alpha(\mathbf{x}_f, t + \delta t) = \frac{1}{1+q} [(1-q) f_\alpha^+(\mathbf{x}_{ff}, t) + q f_\alpha^+(\mathbf{x}_f, t) + q f_\alpha^-(\mathbf{x}_f, t)], \quad (12)$$

where $\bar{\alpha}$ is the opposite direction of α , and q is the fraction of an intersected link in the fluid region, that is written by

$$q = \frac{|\mathbf{x}_f - \mathbf{x}_w|}{|\mathbf{x}_f - \mathbf{x}_b|}, \quad 0 \leq q \leq 1. \quad (13)$$

III. PARALLEL STRATEGY

A. Data parallelism

In the uniform grids, the lattice nodes can be distributed in a simple way among all MPI processes by dividing equally computational domain. The computational domain can be decomposed into sub-domains and distributed to each process in 1D, 2D or 3D ways (See Fig. 4, 5 and 6) in which the fluid domain is decomposed along one, two or three directions.

It can be seen from references [25] the 3D way has the best scalability and efficiency when running on large scale cluster. Therefore, the 3D way is adopted in our parallel strategy. In the 3D way, each MPI process deals with a sub-domain, and

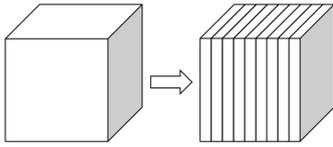


Fig. 4. 1D

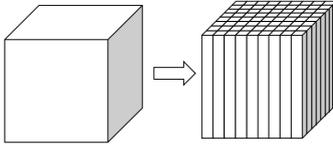


Fig. 5. 2D

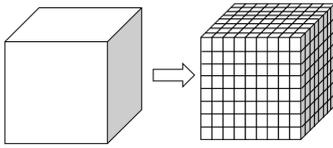


Fig. 6. 3D

the process can be noted as a triple (i, j, k) according to the MPI process rank id . The triple can be obtained by

$$\begin{aligned} i &= \text{mod}(\text{mod}(id, px \times py), px), \\ j &= \frac{\text{mod}(id, px \times py)}{px}, \\ k &= \frac{id}{px \times py}. \end{aligned} \quad (14)$$

where px and py denote the number of domain decomposition along x and y respectively. Then, the computational part along x direction of process id can be got by

$$xBeg = i \times \frac{numX}{px} + \text{minInt}(i, \text{mod}(numX, px)), \quad (15)$$

$$xEnd = \begin{cases} xBeg + \frac{numX}{px} - 1, & \text{mod}(numX, px) \leq i, \\ xBeg + \frac{numX}{px}, & \text{mod}(numX, px) > i. \end{cases} \quad (16)$$

where $numX$ is the lattice node along the x direction. $xBeg$ is the start lattice node along x direction, and $xEnd$ is the end one. The range of y and z directions can be calculated in the same way.

B. Data exchange

Based on Eq. (6) and (8), the only communication occurs in the propagation step. After collision, each MPI process needs to exchange the interface between neighbor MPI processes. In 3D, each MPI process (i, j, k) have to send data of 6 surfaces and 12 edges and receive data of 6 surfaces (Fig. 7) and 12 edges (Fig. 8). When transferring data $\widehat{f}_\alpha(\mathbf{x} - \Delta x_\alpha, t)$ or $\widehat{f}_\alpha(\mathbf{x} + \Delta x_\alpha, t)$, there is no need to transfer all lattice data of \widehat{f}_α , only the expected in propagation operation are required along the direction. For example, Only the data of $\widehat{f}_1, \widehat{f}_7, \widehat{f}_9, \widehat{f}_{11}, \widehat{f}_{13}$ are expected when transferred from MPI process (i, j, k) to $(i + 1, j, k)$.

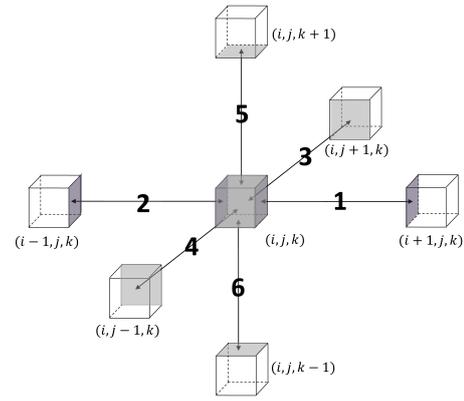


Fig. 7. Surfaces needed to be exchanged

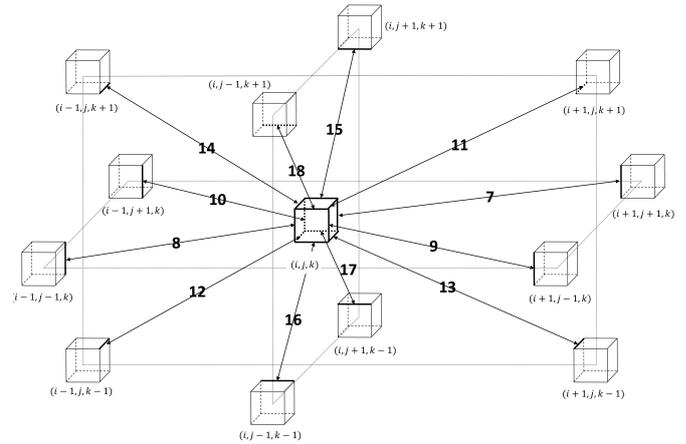


Fig. 8. Edges needed to be exchanged

C. The framework of parallel algorithm of finite difference lattice Boltzmann method

There are two propagation schemes: out-of-place propagation in which the collision step is carried out before the propagation step and in-place propagation in which the collision step is executed after the propagation step [26]. Here out-of-place propagation is taken which is suitable for homogeneous computing environment.

The framework of FDLBM can be described as Algorithm 1. After initializing the flow field information, parallel iterative computation begins in which out-of-place propagation is adopted. Between collision and propagation operation, MPI processes have to exchange data with neighbor processes. When propagation operation is finished, macroscopic quantities and boundary condition should be handled.

IV. NUMERICAL RESULTS

The lid-driven cavity flow (Fig. 9) in 3D is used as a test case to observe the parallel efficiency of FDLBM. The 3D lid-driven flow prescribed by the 3D incompressible Navier-Stokes equation on a square domain $\Omega := (x, y, z) \in [0, L] \times [0, L] \times [0, L] = [0, 1] \times [0, 1] \times [0, 1]$ is

Algorithm 1 Parallel Algorithm of finite difference LBM.

- 1: Initialize the flow field information;
- 2: Parallel iterative computation of collision and propagation until satisfying the convergence condition;
 - (a) Collision operation, as shown by Eq. (6);
 - (b) Exchange data with neighbor MPI processes;
 - (c) Propagation operation through Eq. (8);
 - (d) Calculate the macroscopic variables: density ρ , speed \mathbf{u} , and pressure p based on the lattice nodes using Eq. (5);
 - (e) Deal with boundary conditions.
- 3: Output the result.

$$\begin{cases} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0 \\ \mathbf{u}(x, y, 1) = (U, 0, 0) \\ \mathbf{u}(x, y, 0) = \mathbf{0} \\ \mathbf{u}(0, y, z) = \mathbf{u}(1, y, z) = \mathbf{0} \\ \mathbf{u}(x, 0, z) = \mathbf{u}(x, 1, z) = \mathbf{0} \end{cases} \quad (17)$$

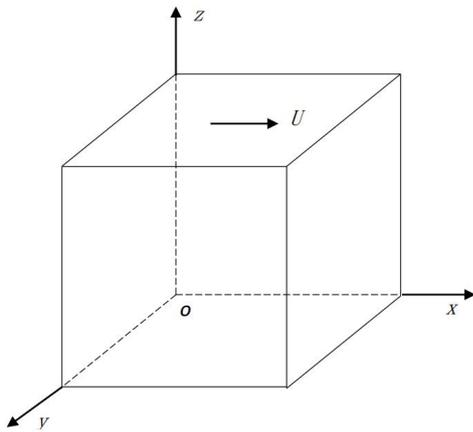


Fig. 9. Sketch map of lid-driven cavity

where $p(\mathbf{x}, t)$ and $\mathbf{u}(\mathbf{x}, t)$ are the pressure and velocity field at site \mathbf{x} and time t respectively. U is the sliding velocity of top wall. The Reynolds number of the flow is defined by U , the dimension L , and the viscosity ν , that is, $Re = \frac{UL}{\nu}$.

A. Numerical validation

In our experiment, the lattice scale is $192 \times 192 \times 192$, the flow with $Re = 1000$ are simulated. The criterion for reaching steady state in the simulations is given by

$$\frac{\sum_i \|\mathbf{u}(\mathbf{x}_i, t + \delta t) - \mathbf{u}(\mathbf{x}_i, t_n)\|_2}{\sum_i \|\mathbf{u}(\mathbf{x}_i, t_n)\|_2} < 10^{-6}, \quad (18)$$

where $\|\mathbf{u}\|_2$ represents the L^2 norm of \mathbf{u} .

Fig. 10 shows the vortical structure of 3D cavity flow, Fig. 11 and Fig.12 show the streamlines of x-z and y-z planes respectively, and Fig. 13 shows the value of u along the line of $x = 0.5$ and $y = 0.5$. The simulation results agree well with reference [27].

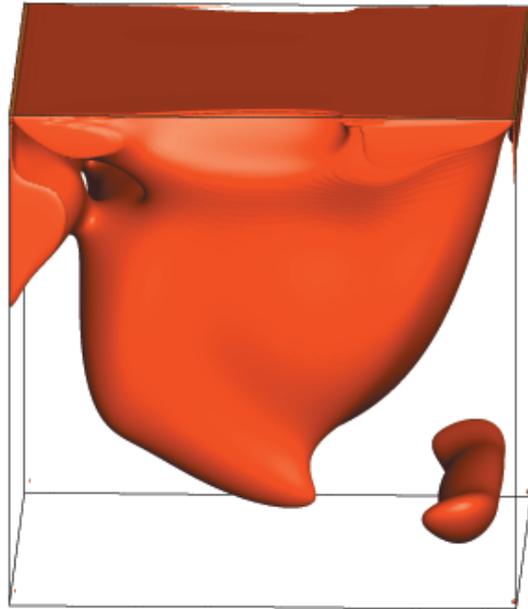


Fig. 10. Vortical structure of 3D cavity flow

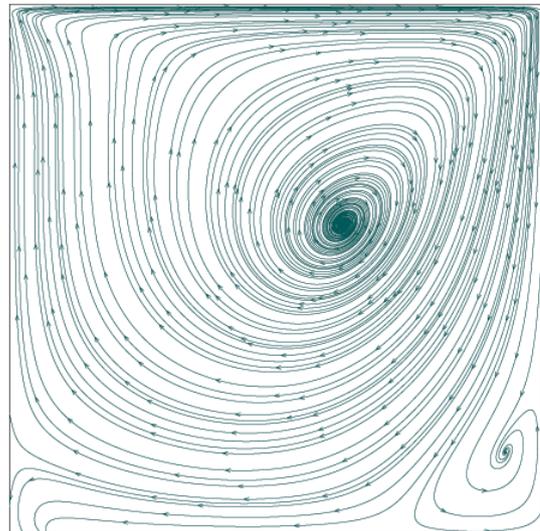


Fig. 11. Streamlines of x-z plane ($y=0.5$)

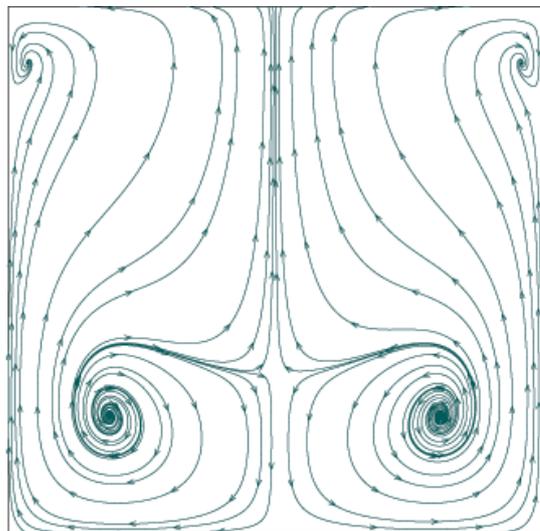


Fig. 12. Streamlines of y-z plane ($x=0.5$)

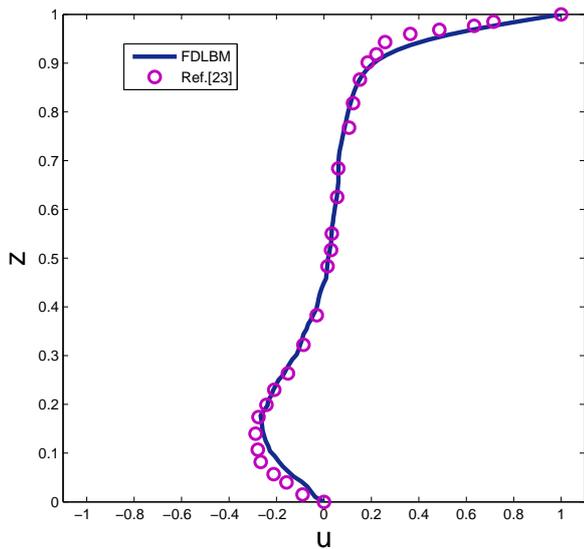


Fig. 13. $u(x=0.5, y=0.5)$

B. Parallel performance

The performance of our parallel algorithm is addressed on Sunway Blue Light MPP supercomputer in National Supercomputing Center in Jinan (NSCC-JN). The supercomputer is completely built with Chinese-designed and manufactured Shenwei processors. It is equipped with 8,700 ShenWei SW1600 processors and each processor has sixteen cores, 1.0-1.1GHz, 128 Gflops, and 16GB RAM. Its peak performance is 1.07 Pflops/s and LINPACK efficiency is 74.37% with the sustained performance of 795.9 Tflop/s.

The test problem is the lid driven cavity flow at Reynolds number being 1000. Each test case executes 1000 iterations. The grid size with $128 \times 128 \times 128$, $192 \times 192 \times 192$, $256 \times 256 \times 256$, $384 \times 384 \times 384$ is tested respectively.

Figure 14 and Figure 15 show the comparison of speedup and efficiency among different grid size respectively. It can be seen that the larger the grid size is, the better the speedup and efficiency are. The speedup and efficiency improve when the grid size increases. when the grid size is $384 \times 384 \times 384$ and the cores are 3072, the efficiency can achieve 86.1%.

In order to compare the efficiency with the same scale (8192 lattices) on each core, the grid size of $128 \times 128 \times 128$, $192 \times 192 \times 192$ and $384 \times 384 \times 384$ are executed on 256(8, 8, 6), 1296 (12, 12, 9) and 3072(16, 16, 12) cores respectively. Figure 16 denotes the efficiency comparison. When the cores increase, the presented algorithm has a good scalability.

Table I give the computation time and communication time among different cores. As the cores increase, the communication time decreases.

C. Flow past a sphere

In this subsection, a 3D incompressible flow around a sphere with a constant velocity profile, $u = U_\infty = \{0.1Ma, 0, 0\}$, was simulated as a numerical example. The sphere radius $D = 20$ are taken. Figure 17 shows the flow geometry, coordinate system, and computational domain. In order to eliminate the effect of boundaries, the length of

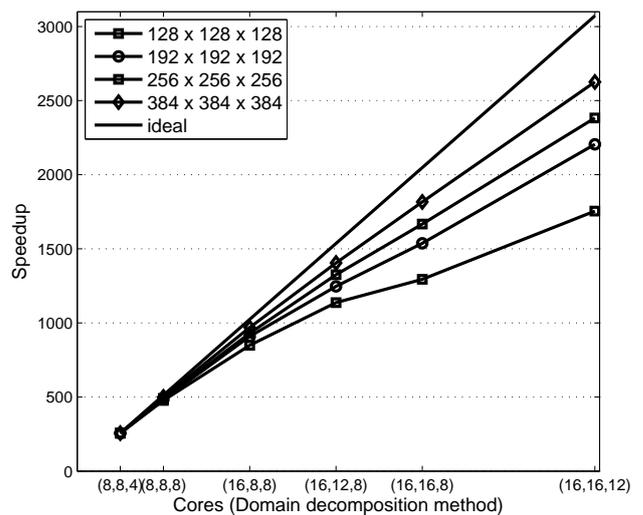


Fig. 14. Speedup comparison of difference scale

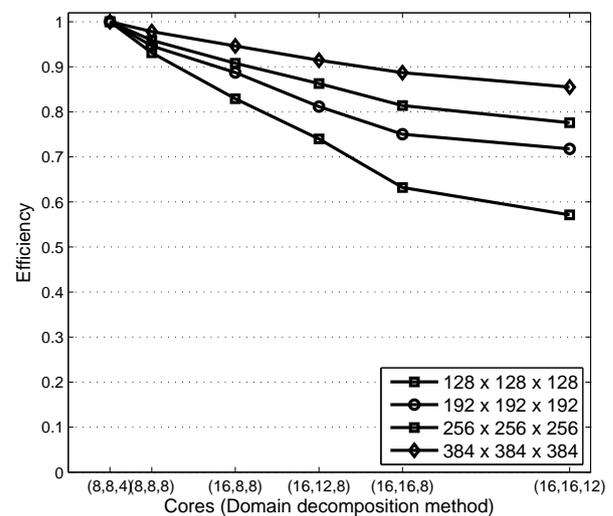


Fig. 15. Efficiency comparison of difference scale

TABLE I
COMPUTATION AND COMMUNICATION TIME WHEN GRID SIZE IS 384×384

Cores	computation time(s)	communication time(s)
3072 (16, 16, 12)	465.893	7.93875
2048 (16, 16, 8)	676.692	8.34563
1536 (16, 12, 8)	877.413	8.52875
1024 (16, 8, 8)	1276.91	7.31812
512 (8, 8, 8)	2474.9	9.28844
256 (8, 8, 4)	4828.46	32.6103

the computational domain is $51.2D$, the width is $12.8D$ and the height is $12.8D$. The lattice scale is $1024 \times 256 \times 256$. Figure 18 and Figure 19 show the streamlines of the simulation result when $Re=100$ and $Re=200$ respectively which are feasible and correct.

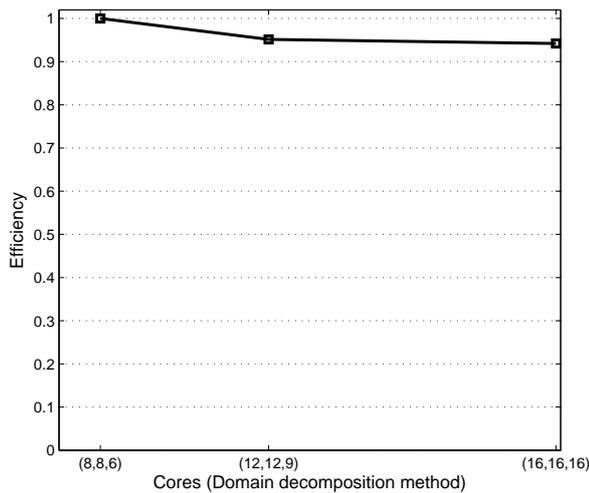


Fig. 16. Efficiency comparison of same scale



Fig. 17. Flow geometry, coordinate system, and computational domain

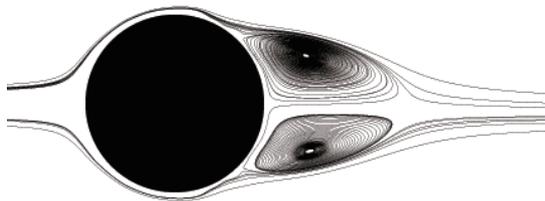


Fig. 18. Streamlines of a slice of simulation result (Re=100)

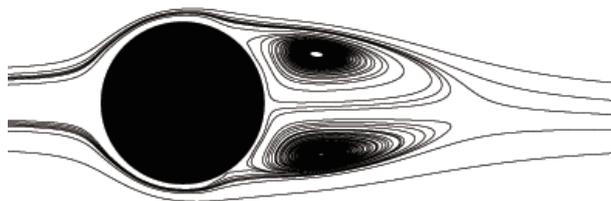


Fig. 19. Streamlines of a slice of simulation result (Re=200)

V. CONCLUSIONS

The present work has studied the parallel algorithm of FDLBM including domain decomposition method and data exchange strategy. The only communication occurs after collision during executing entire algorithm. It is a best choice to choose the 3D DDM, because others are not suitable for large scale cluster. According to the numerical results in the cluster of NSCC-JN, the presented algorithm are efficient and scalable.

ACKNOWLEDGMENT

The authors thank Dr. Meng Guo in the NSCC-JN for his good advice and help. The authors are also grateful to the reviewers for their thoughtful comments and valuable suggestions. This research was supported by the Major Research Plan of NSFC [No. 91330116] and [No. 91630206].

REFERENCES

- [1] N. Meghanathan and G. W. Skelton, "Risk Notification Message Dissemination Protocol for Energy Efficient Broadcast in Vehicular Ad hoc Networks", *IAENG International Journal of Computer Science*, vol. 37, no. 1, pp. 1-10, 2010.
- [2] A. Dadvand, M. Baghalnezhad, I. Mirzaee, B. C. Khoo and S. Ghor-eishi, "An immersed boundary-lattice boltzmann approach to study the dynamics of elastic membranes in viscous shear flows", *Journal of Computational Science*, vol. 5, no. 5, pp. 709-718, Sep. 2014.
- [3] L. AJC, "Lattice-Boltzmann methods for suspensions of solid particles", *Molecular Physics*, vol. 113, no. 17-18, pp. 2531-2537, Sep. 2015.
- [4] W. Wang and J.-G. Zhou, "Lattice Boltzmann method for axisymmetric turbulent flows", *International Journal of Modern Physics C*, vol. 26, no. 9, pp. 54-67, Sep. 2015.
- [5] Y. H. Qian, D. D'Humières and P. Lallemand, "Lattice BGK models for Navier-Stokes equation", *Europhysics Letters*, vol. 17, no. 6, pp. 479-484, Feb. 1992.
- [6] S. Chen, H. Chen, D. Martnez and W. Matthaeus, "Lattice Boltzmann model for simulation of magnetohydrodynamics", *Physical Review Letters*, vol. 67, no. 27, pp. 3776-3779, Dec. 1991.
- [7] S. Ansumali and I. V. Karlin, "Entropy Function Approach to the Lattice Boltzmann Method", *Journal of Statistical Physics*, vol. 107, no. 1-2, pp. 291-308, Dec. 2002.
- [8] I. Ginzburg, F. Verhaeghe and D. d'Humières, "Two-relaxation-time lattice Boltzmann scheme: about parametrization, velocity", *Communications in Computational Physics*, vol. 3, no. 2, pp. 291-308, Feb. 2008.
- [9] Pierre Lallemand and Li-Shi Luo, "Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability", *Physical Review E*, vol. 61, no. 26, pp. 6546-6562, Jan. 2000.
- [10] N. Cao, S.-Y. Chen, S. Jin and D. Martinez, "Physical symmetry and lattice symmetry in the lattice Boltzmann method", *Physical Review E*, vol. 55, no. 1, pp. R21-R24, Jan. 1997.
- [11] Z. L. Guo and T. S. Zhao, "Explicit finite-difference lattice Boltzmann method for curvilinear coordinates", *Physical Review E*, vol. 67, no. 6, pp. 066709, Jun. 2003.
- [12] T. Lee and C.-L. Lin, "A characteristic Galerkin method for discrete Boltzmann equation", *Journal of Computational Physics*, vol. 171, no. 1, pp. 336-356, Jul. 2001.
- [13] F. Nannelli and S. Succi, "The lattice Boltzmann-equation on irregular lattices", *Journal of Statistical Physics*, vol. 68, no. 3-4, pp. 401-407, Aug. 1992.
- [14] D. V. Patila, K. N. Lakshmish, "Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh", *Journal of Computational Physics*, vol. 228, no. 14, pp. 5262-5279, Aug. 2009.
- [15] T. Lee, C.-L. Lin, "An Eulerian description of the streaming process in the lattice Boltzmann equation", *Journal of Computational Physics*, vol. 185, no. 2, pp. 445-471, Mar. 2003.
- [16] A. Fakhari and T. Lee, "Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique", *Physical Review E*, vol. 89, no. 3, pp. 033310, Mar. 2014.
- [17] A. Fakhari and T. Lee, "Numerics of the lattice boltzmann method on nonuniform grids: Standard LBM and finite-difference LBM", *Computers & Fluids*, vol. 107, pp. 205-213, Jan. 2015.
- [18] J. V. Tembhurne and S. R. Sathe, "Multiple Precision Integer GCD Performance Analysis on Parallel Architectures", *IAENG International Journal of Computer Science*, vol. 42, no. 4, pp. 296-304, 2015.
- [19] H.-Y. Li, J. Teng, Z.-H. Li and L. Zhang, "Nonlinear Dynamic Analysis Efficiency by Using a GPU Parallelization", *Engineering Letters*, vol. 23, no. 4, pp. 232-238, 2015.
- [20] L.-S. Luo, W. Liao, X.-W. Chen, Y. Peng and W. Zhang, "Numerics of the lattice boltzmann method: Effects of collision models on the lattice boltzmann simulations", *Physical Review E*, vol. 83, no. 5, pp. 056710, May 2011.
- [21] V. Sofonea and R. F. Sekerka, "Viscosity of finite difference lattice Boltzmann models", *Journal of Computational Physics*, vol. 184, no. 2, pp. 422-434, Jan. 2003.
- [22] Z. Guo, C. Zheng and B. Shi, "Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method", *Chinese Physics*, vol. 11, no. 4, pp. 365-374, Apr. 2002.

- [23] D. Yu, R. Mei, W. Shyy, "A Unified Boundary Treatment in Lattice Boltzmann Method", 41st Aerospace Sciences Meeting and Exhibit, (2003)
- [24] C. Schepke, N. Maillard and P. O. A.Navaux, "Parallel Lattice Boltzmann Method with Blocked Partitioning", *International Journal of Parallel Programming*, vol. 37, no. 6, pp. 593-611, Dec. 2009.
- [25] X. Wang and A. Takayuki, "Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster", *Parallel Computing*, vol. 37, no. 9, pp. 521-535, Sep. 2011.
- [26] C. Obrecht, F. Kuznik, B. Tourancheau and J.-J. Roux, "The TheLMA project: Multi-GPU implementation of the lattice Boltzmann method", *International Journal of High Performance Computing Applications*, vol. 25, no. 3, pp. 521-535, Aug. 2011.
- [27] R. Iwatsu, J. M. Hyun and K. Kuwahara, "Analyses of three dimensional flow calculations in a driven cavity", *Fluid Dynamics Research*, vol. 6, no. 2, pp. 91-102, Jul. 1990.